

Status

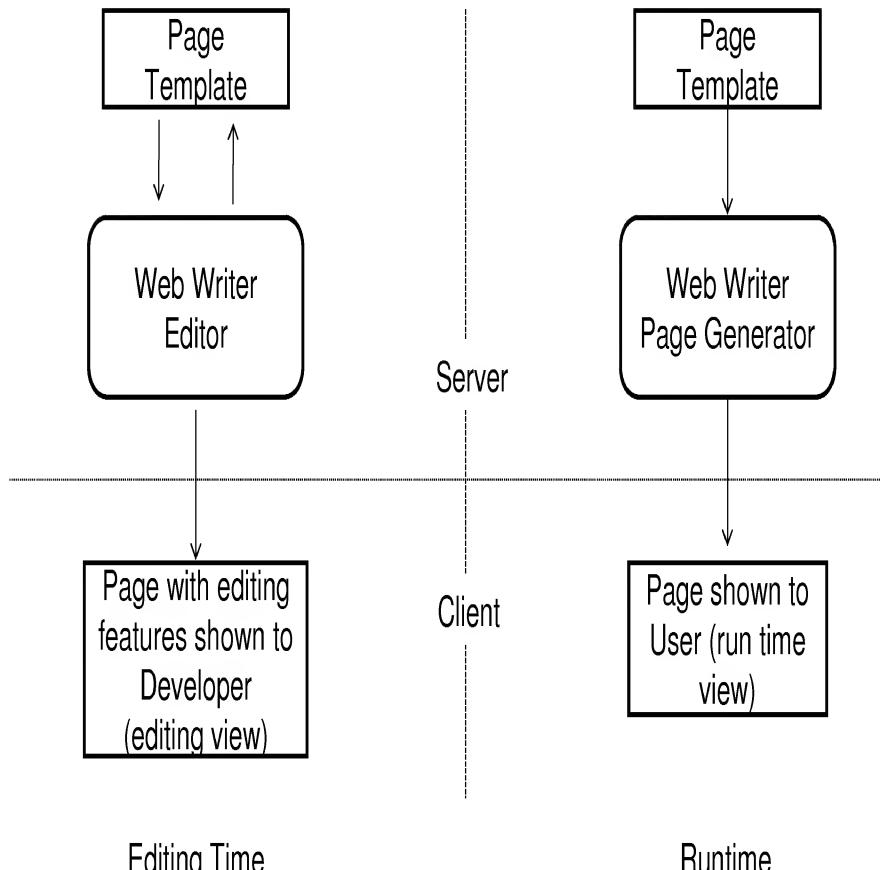
- Claims rejected mainly on WebWriter I article
- Today's objective:
Resolve Disagreement in Interpretation of Prior Art
- WebWriter II article, discussed later
- Section 101 issues, discussed at end

Background – WebWriter I

- A Web / Browser Based Editor
- Editing of *Dynamic Web Page Templates* / *Dynamic Web Applications*

- Page Templates have Dynamic Areas
- Content of Dynamic Areas is computed dynamically at a page request

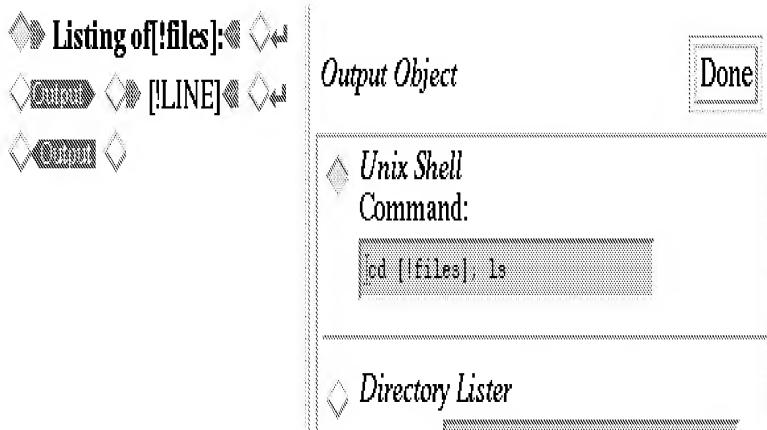
Overview



Distinctive Feature(s)

- WebWriter shows Placeholders for Dynamic Areas during editing
 - Editing view and runtime view differ
- Applicant's editor shows computed Content during editing

See yourself: The edited template includes placeholders for dynamic items, as shown in Fig. 7b



Editing-View in WebWriter

For Comparision run-time view as shown in Fig. 6b

Listing of /project/girweb/figs/coins/:
coins.htm
dime.gif
nickel.gif
penny.gif
quarter.gif

Run-time view in WebWriter

Citation (1) from WebWriter

- “*static parts of each page are created in advance by the designer using a text editor*” (see p. 2, 2nd ¶, under heading “The WebWriter Application Model”)
- “*WebWriter then adds a placeholder for the output area [which] will be replaced **at runtime** by the output of a script . . .*” (see p. 4, 2nd ¶, under heading “Placing Dynamic Areas”, emphasis added).

- WebWriter uses placeholders
- WebWriters dynamic areas have scripts
- scripts run at **runtime**

Citation (2) from WebWriter

- “**Once** a Web Writer application is created and **saved** to disk it can be **run**” p. 6, col. 2 of WebWriter I, in the section entitled “Running the Application” (emphasis added)
- application is run only after being saved i.e. after editing

Distinctive Feature(s)

- WebWriter shows Placeholders for Dynamic Areas during editing
 - Applicant's editor shows computed Content
- WebWriter does not run the scripts of dynamic areas during editing
 - Applicant's editor runs the edited application during editing

Examiner's Response to Arguments

(See OA @ p. 23)

- Misinterpreted Limitations:

“Running saved content” - but Distinctive feature is

“Running unsaved content”, “Running before saving”

“Running during editing”

“only running during editing” - but Distinctive feature is

“Running during editing”

- “Examiner believes that WebWriter does in fact disclose *this limitation* in his disclosure and that applicant’s claim doesn’t include or exclude any of the teachings”

Examiners Response to Argument (p 24)

“it is actually quite inherent that the button is an executable component and hence this is being performed during editing” (see p. 24)

Applicant disagrees. The fact that the button is executable does not give any indication as to when it is performed, e.g. during editing or at runtime.

WebWriter Discloses

two **separate** programs

WebWriter Editor

- for Editing
- for Use at Editing Time
- Generates Pages with Editing Features and Placeholders

WebWriter Page Generator

- for Viewing Pages
- for Use at Run Time, i.e. after editing and not during editing
- Generates final pages by executing scripts of dynamic areas

Citation (3)

“WebWriter includes a direct manipulation Web page editor, the **WebWriter Editor**, which runs in the web browser as a CGI-service, and the **WebWriter Page generator**, which creates new pages as the application runs” (See abstract, 2nd para.)

- indicates that the WebWriter Editor and the WebWriter Page generator are two separate programs

Citation (4)

- “**When** a WebWriter application is **running**, each new page is assembled by the **WebWriter Page Generator**, **another** server based CGI C++ **program...**”

first paragraph of the “WebWriter Page Generator” section on page 9 (emphasis added) .

- “another” means it is a separate program
- “when running” indicates that there is a time when it is not running
- it is the page generator that runs the application and not the editor

Examiner's Response to Argument (p. 24)

“WebWriter includes a direct manipulation Web page editor, the **WebWriter Editor**, which runs in the web browser as a CGI-service, and the **WebWriter Page Generator**, which creates new pages as the application runs” 2nd paragraph of the web writer abstract

The Examiner reasons:

“the WebWriter Page Generator runs the Application”

Applicant says:

This does not state anything about the WebWriter Editor, only about the WebWriter Page Generator, which is a separate program running at a different time.

Applicant's Claims require an Editor that runs part of the application, but the WebWriter Editor does not.

Examiner's Response to Argument (p. 24)

- Examiner: "Web Writer is said to be an editor that runs in a web browser which seamlessly generates/creates web pages while the application is run" (citing p. 4 under the heading "Creating the Template Pages")
- The cited portion only talks about running the WebWriter Editor not about running the application being edited.

WebWriter II

- Cited on p. 4 of OA for claim 1 because “WebWriter does not disclose having a server” and for dependent claims
- Not just an improved version but new program with a totally different architecture (called meteor shower) [Sec 1.2 p 1508]
 - therefore combination not possible
- Performs most of the editing operations on the client computer (except for loading the document before editing and saving the document after editing) [page 1511, 3rd paragraph “The CGI modules provide server-side services such as loading files, parsing HTML, saving files, setting up the environment for the HTML modules at start-up. The JavaScript modules handle displaying the edited page ...”]
- Teaches away from running application on server during editing, because changes are transferred to the server on save and not before (as it would be required for running a server side application during editing)

Claims 1, 22, 125

- Examiner: plain language of claims as disclosed in claim 1, 6 and 22, doesn't preclude, include or exclude when execution or running of the editor as taught by Web Writer is performed.
- Claim 1 requires "the applications generating generated documents" ... "said generated documents including additional editing features"
 - Thus, editing and running take place simultaneously

Examiner reasoning about claim language

- Examiner: “Running saved content such as a template on the fly, in one or more steps isn't disclosed as not meeting Applicant's claimed limitations as recited in Applicant's claims.”
- Applicant: The point is about content that is not (yet) saved, i.e. during editing. Claims require running during editing, which is not disclosed by WebWriter

Claim 22 and 59

- Claim 22: “an editor program operating within the web browser on generated documents” ... “a document generator program having instructions ... for **executing components**, and for generating **the** generated documents from the document templates”
- Claim 59: “a document generator program having instructions for generating generated documents from dynamic web documents which look and function similar to the end user’s view of the documents, the editor program comprising first instructions for requesting that the document generator program processes a dynamic web document during editing thereby resulting in a generated document”

Claim 6, 26

- Directly require similarity of runtime view and editing view: Claim 6 “...whereon the component has a similar appearance as on the generated page with the addition of editing features.”
- Claim 26 “...part of the second document appears and functions similar to the run-time view of the first document”

✓ As seen above WebWriter shows placeholders that do not appear similar to the runtime content

Claim 90, 125

- Directly require running during editing
- Claim 90: “scripts contained in said document remain running during editing”
- Claim 125: “A method for dynamically editing an application ... comprising the steps of running at least part of the application, thereby executing selected components”

Claim 51, 74, 114

- Claim 51 “a plurality of components containing instructions to generate browser code each component having features to cooperate in editing the component”
- Claim 74 “a plurality of components having instructions to generate browser code, ... the components having the ability to cooperate with the editor”
- Similar language in Claim 114. (“a plurality of components having instructions for execution on the server computer, at least one of the components including first features adapted to cooperate with an editor in editing said component”)
- WebWriters components are scripts and the only operation on scripts described is running. Running takes place however at runtime only in WebWriter. So there is no cooperation of components and editor in WebWriter.

Conclusions

- WebWriter does not run the application during editing
- Section 101 issues are resolved ?
- How best to proceed ?